**Лабораторное занятие № 4.**

**Сетевые протоколы**

**Лабораторное (практическое) занятие** рассчитано на 2 часа, относится к теме «Общие принципы построения сетей»

**Формируемые компетенции:** ОК.1-9 ПК 2.3 - 2.4

**Цель:** Изучить сетевые протоколы передачи информации.

**Методическое и техническое обеспечение:** IBM PC-совместимый компьютер.

**Протокол SMTP**

Главной целью протокола **simple mail transfer protocol** (**SMTP**) служит надежная и эффективная доставка электронных почтовых сообщений. SMTP является довольно независимой субсистемой и требует только надежного канала связи. Средой для SMTP может служить отдельная локальная сеть, система сетей или весь Интернет.

**SMTP** базируется на следующей модели коммуникаций: в ответ на запрос пользователя почтовая программа-отправитель устанавливает двухстороннюю связь с программой-приемником (TCP). Получателем может быть оконечный или промежуточный адресат. **SMTP**-команды генерируются отправителем и посылаются получателю. На каждую команду должен быть отправлен и получен отклик.

Когда канал организован, отправитель посылает команду **MAIL**, идентифицирую себя. Если получатель готов к приему сообщения, он посылает положительное подтверждение. Далее отправитель посылает команду RCPT, идентифицируя получателя почтового сообщения. Если получатель может принять сообщение для оконечного адресата, он выдает снова положительное подтверждение. В противном случае он отвергает получение сообщения для данного адресата, но не вообще почтовой посылки. Взаимодействие с почтовым сервером возможно и в диалоговом режиме.

**SMTP** — требующий соединения текстовый протокол, по которому отправитель сообщения связывается с получателем посредством выдачи командных строк и получения необходимых данных через надёжный канал, в роли которого обычно выступает TCP-соединение (Transmission Control Protocol — протокол управления передачей). **SMTP**-сессия состоит из команд, посылаемых **SMTP**-[клиентом](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), и соответствующих ответов **SMTP**-[сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)). Когда сессия открыта, сервер и клиент обмениваются её параметрами. Сессия может включать ноль и более **SMTP**-операций (транзакций).

**SMTP**-операция состоит из трёх последовательностей команда/ответ. Описание последовательностей:

* **MAIL FROM** — устанавливает обратный адрес (т. е. Return-Path, 5321.From, mfrom). Это адрес для [возвращённых писем](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%B7%D0%B2%D1%80%D0%B0%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D0%B8%D1%81%D1%8C%D0%BC%D0%BE).
* **RCPT TO** — устанавливает получателя данного сообщения. Эта команда может быть дана несколько раз, по одному на каждого получателя. Эти адреса также являются частью оболочки.
* **DATA** — для отправки текста сообщения. Это само содержимое письма, в противоположность его оболочке. Он состоит из заголовка сообщения и тела сообщения, разделенных пустой строкой. DATA, по сути, является группой команд, а сервер отвечает дважды: первый раз на саму команду DATA, для уведомления о готовности принять текст; и второй раз после конца последовательности данных, чтобы принять или отклонить всё письмо.

Помимо промежуточных ответов для DATA-команды, каждый ответ сервера может быть положительным (код ответа 2хх) или отрицательным. Последний, в свою очередь, может быть постоянным (код 5хх) либо временным (код 4хх). Отказ SMTP-сервера в передаче сообщения — постоянная ошибка; в этом случае клиент должен отправить возвращённое письмо. После сброса — положительного ответа, сообщение скорее всего будет отвержено. Также сервер может сообщить о том, что ожидаются дополнительные данные от клиента (код 3xx).

### SMTP и извлечение сообщений

**SMTP** — всего лишь протокол доставки. Он не может по требованию взять сообщения с удаленного сервера. Для извлечения почты и управления почтовым ящиком разработаны другие протоколы, такие как **POP** и **IMAP**. Тем не менее, **SMTP** предоставляет возможность начать на удаленном сервере обработку очереди сообщений, при которой запрашивающая система может получать все направленные ей сообщения. **POP** и **IMAP** предпочтительны, когда компьютер пользователя включен не постоянно, или же временно подключен к Интернету.

**Протокол POP3**

**POP3** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Post Office Protocol Version 3* — [протокол](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) почтового отделения, версия 3) — стандартный [интернет-протокол прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F), используемый [клиентами электронной почты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) для получения почты с удаленного [сервера](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) по [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP)-соединению.

POP и [IMAP](https://ru.wikipedia.org/wiki/IMAP) (Internet Message Access Protocol) — наиболее распространенные [интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)-протоколы для извлечения почты. Практически все современные клиенты и серверы электронной почты поддерживают оба стандарта. Протокол POP был разработан в нескольких версиях, нынешним стандартом является третья версия (POP3). Большинство [поставщиков услуг электронной почты](https://ru.wikipedia.org/wiki/Webmail) (такие как [Hotmail](https://ru.wikipedia.org/wiki/Hotmail),[Gmail](https://ru.wikipedia.org/wiki/Gmail) и Yahoo! Mail) также поддерживают IMAP и POP3. Предыдущие версии протокола (POP, POP2) устарели.

POP поддерживает простые требования «загрузи-и-удали» для доступа к удаленным почтовым ящикам. Хотя большая часть POP-клиентов предоставляет возможность оставить почту на сервере после загрузки, использующие POP клиенты обычно соединяются, извлекают все письма, сохраняют их на пользовательском компьютере как новые сообщения, удаляют их с сервера, после чего разъединяются.

Другие протоколы, в частности IMAP, предоставляют более полный и комплексный удаленный доступ к типичным операциям с почтовым ящиком. Многие клиенты электронной почты поддерживают как POP, так и IMAP; однако, гораздо меньше [интернет-провайдеров](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D1%80%D0%BE%D0%B2%D0%B0%D0%B9%D0%B4%D0%B5%D1%80) поддерживают IMAP.

POP3-сервер прослушивает [общеизвестный порт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BF%D0%BE%D1%80%D1%82%D0%BE%D0%B2_TCP_%D0%B8_UDP) 110. [Шифрование](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) связи для POP3 запрашивается после запуска протокола, с помощью либо команды STLS (если она поддерживается), либо POP3S, которая соединяется с сервером используя [TLS](https://ru.wikipedia.org/wiki/TLS) или [SSL](https://ru.wikipedia.org/wiki/SSL) по TCP-порту 995.

Доступные сообщения клиента фиксируются при открытии почтового ящика POP-сессией и определяются количеством сообщений для сессии, или, по желанию, с помощью уникального идентификатора, присваиваемого сообщению POP-сервером. Этот уникальный идентификатор является постоянным и уникальным для почтового ящика и позволяет клиенту получить доступ к одному и тому же сообщению в разных POP-сессиях. Почта извлекается и помечается для удаления с помощью номера сообщения. При выходе клиента из сессии помеченные сообщения удаляются из почтового ящика.

**Протокол HTTP**

**HTTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *HyperText Transfer Protocol* — «протокол передачи [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82)») — [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) передачи данных (изначально — в виде гипертекстовых документов в формате [HTML](https://ru.wikipedia.org/wiki/HTML), в настоящий момент используется для передачи произвольных данных). Основой HTTP является [технология «клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80), то есть предполагается существование потребителей ([клиентов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9))), которые инициируют соединение и посылают [запрос](https://ru.wikipedia.org/w/index.php?title=HTTP-%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81&action=edit&redlink=1), и поставщиков ([серверов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5))), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Основным объектом манипуляции в HTTP является ресурс, на который указывает [URI](https://ru.wikipedia.org/wiki/URI) (Universal Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере [файлы](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и [ответе](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D0%B0#.D0.92.D0.B7.D0.B0.D0.B8.D0.BC.D0.BE.D0.B4.D0.B5.D0.B9.D1.81.D1.82.D0.B2.D0.B8.D0.B5_.D1.81_.D0.B2.D0.B5.D0.B1-.D1.81.D0.B5.D1.80.D0.B2.D0.B5.D1.80.D0.BE.D0.BC_.28HTTP.29) способ представления одного и того же ресурса по различным параметрам: формату, [кодировке](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B1%D0%BE%D1%80_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%BE%D0%B2), языку и т. д. Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

HTTP — протокол [прикладного уровня](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI), аналогичными ему являются [FTP](https://ru.wikipedia.org/wiki/File_Transfer_Protocol) и [SMTP](https://ru.wikipedia.org/wiki/SMTP). Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные [URI](https://ru.wikipedia.org/wiki/URI). В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ». Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами. Браузер, посылающий запросы, может отслеживать задержки ответов. Сервер может хранить [IP-адреса](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) и заголовки запросов последних клиентов. Однако сам протокол не осведомлён о предыдущих запросах и ответах, в нём не предусмотрена внутренняя поддержка состояния, к нему не предъявляются такие требования.

Всё программное обеспечение для работы с протоколом HTTP разделяется на три большие категории:

* Серверы как основные поставщики услуг хранения и обработки информации (обработка запросов).
* Клиенты — конечные потребители услуг сервера (отправка запроса).
* Прокси для выполнения транспортных служб.

Для отличия конечных серверов от прокси в официальной документации используется термин «исходный сервер» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *origin server*). Один и тот же программный продукт может одновременно выполнять функции клиента, сервера или посредника в зависимости от поставленных задач. В спецификациях протокола HTTP подробно описывается поведение для каждой из этих ролей.

**Протокол FTP**

**FTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *File Transfer Protocol* — [протокол передачи](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) файлов) — стандартный [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), предназначенный для передачи файлов по TCP-сетям (например, Интернет). FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3).

Протокол построен на архитектуре «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль [открытым текстом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол [SSH](https://ru.wikipedia.org/wiki/SSH) для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

Первые клиентские FTP-приложения были интерактивными инструментами командной строки, реализующими стандартные команды и синтаксис. [Графические пользовательские интерфейсы](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) с тех пор были разработаны для многих используемых по сей день операционных систем. Среди этих интерфейсов как программы общего веб-дизайна вроде [Microsoft Expression Web](https://ru.wikipedia.org/wiki/Microsoft_Expression_Web), так и специализированные FTP-клиенты (например, FileZilla).

### Отличие от HTTP

Достаточно яркая особенность протокола FTP в том, что он использует множественное (как минимум — двойное) подключение. При этом один канал является управляющим, через который поступают команды серверу и возвращаются его ответы (обычно через TCP-порт 21), а через остальные происходит собственно передача данных, по одному каналу на каждую передачу. Поэтому в рамках одной сессии по протоколу FTP можно передавать одновременно несколько файлов, причём в обоих направлениях. Для каждого канала данных открывается свой TCP порт, номер которого выбирается либо сервером, либо клиентом, в зависимости от режима передачи.

Протокол FTP имеет двоичный режим передачи, что сокращает накладные расходы [трафика](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%82%D1%80%D0%B0%D1%84%D0%B8%D0%BA) и уменьшает время обмена данными при передаче больших файлов. Протокол же HTTP обязательно требует кодирования двоичной информации в текстовую форму, например при помощи алгоритма [Base64](https://ru.wikipedia.org/wiki/Base64).

Начиная работу через протокол FTP, клиент входит в сессию, и все операции проводятся в рамках этой сессии (проще говоря, сервер помнит текущее состояние). Протокол HTTP ничего не «помнит» — его задача — отдать данные и забыть, поэтому запоминание состояния при использовании HTTP осуществляется внешними по отношению к протоколу методами.

FTP работает на [прикладном уровне](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) [модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI) и используется для передачи файлов с помощью [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP). Для этого должен быть запущен FTP-сервер, ожидающий входящих запросов. [Компьютер-клиент](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) может связаться с сервером по порту 21. Это соединение (поток управления) остаётся открытым на время сессии. Второе соединение (поток данных), может быть открыт как сервером из порта 20 к порту соответствующего клиента (активный режим), или же клиентом из любого порта к порту соответствующего сервера (пассивный режим), что необходимо для передачи файла данных. Поток управления используется для работы с сессией — например, обмен между клиентом и сервером командами и паролями с помощью telnet-подобного протокола.

**Содержание отчета**

Отчет должен содержать следующие пункты:

1. Ответы на контрольные вопросы.

2. Защита работы.

Пример оформление отчета представлен в документе «Пример отчета.doc», который й располагается в папке с заданием.

**Контрольные вопросы**

1. Протокол SMTP?

2. За что отвечают поля MAIL FROM, DATA?

3. Протокол POP3?

4. Протокол HTTP?

5. Протокол FTP?

6. Отличие протоколов FTP и HTTP?

7. Отличие протоколов FTP и SMTP?

**Список литературы:**

1. Баженова И.Ю. Основы проектирования приложений баз данных [Электронный ресурс]: ИНТУИТ, 2017 URL: http://www.iprbookshop.ru/67380.html

2. Л.В.Воробьев, А.В.Давыдов, Л.П.Щербина, Системы и сети передачи информации,Москва ACADEMIA

3. Компьютерные сети [Электронный источник] - http://net.e-publish.ru/p216aa1.html

4. Информатика как наука [Электронный источник] - http://www.inf1.info/book/export/html/122

5. Компьютерные сети и технологии [Электронный источник] - http://www.xnets.ru/